`— title: “ExamPrep” output: pdf\_document —

От упражнение 1

#1.  
#create a vector  
vect <- c(8,3,8,7,15,9,12,4,9,10,5,1)  
#create a 4x3 matrix  
m <- matrix(vect, nrow = 4, ncol = 3)  
#adding a column  
m1 <- cbind(m, c(1,3,5,7))  
  
#indexes of the first column  
ordered <- order(m1[,1], decreasing = FALSE)  
#ordered matrix  
ordered\_by\_first\_column <- m1[ordered,]  
  
#indexes of the first two columns  
ordered2 <- order(m1[,1], m1[,2], decreasing = FALSE)  
#ordered matrix  
ordered\_by\_two\_columns <- m1[ordered2, ]

#2.  
#most and least expensive in 2000  
most <- which.max(homedata$y2000)  
least <- which.min(homedata$y2000)  
  
#prices in 1970  
homedata$y1970[most]

## [1] 198900

homedata$y1970[least]

## [1] 10000

#top five most expensive houses in 2000  
ordered\_five <- homedata$y2000[order(homedata$y2000, decreasing = T)]  
  
top\_five <- head(ordered\_five ,5)  
  
#средната цена на 5те най-скъпи от 2000, но на техните цени от 1970  
mean\_top\_five <- mean(head(homedata$y1970[order(homedata$y2000, decreasing = T)], 5))  
  
#къщите, чийто цена е намаляла през 2000г.  
lowered <- homedata$y2000[which(homedata$y2000 < homedata$y1970)]  
  
percent\_increase <- head(order(((homedata$y2000 - homedata$y1970) / homedata$y1970),   
 decreasing = T), 10)  
  
top\_ten\_increase <- homedata$y2000[percent\_increase]

#3.  
#number of men  
num\_men <- nrow(survey[survey$Sex == 'Male', ])  
  
#number of men smokers  
num\_men\_smokers <- nrow(survey[survey$Sex == 'Male' & survey$Smoke != 'Never', ])  
  
#mean height of all men  
mean(survey$Height[survey$Sex == 'Male'], na.rm = T)

## [1] 178.826

#height and sex of the top 6 youngest students  
youngest <- head(order(survey$Age), 6)  
  
survey$Sex[youngest]

## [1] Male Male Female Female Female Female  
## Levels: Female Male

survey$Height[youngest]

## [1] NA NA NA 160.00 172.00 170.18

От упражнение 2

#1.  
#случайно избран човек да се окаже пушач  
survey$Smoke %>% table() %>% prop.table()

## .  
## Heavy Never Occas Regul   
## 0.04661017 0.80084746 0.08050847 0.07203390

#случайно избран мъж да се окаже редовно пушещ  
table(survey$Smoke, survey$Sex) %>% prop.table()

##   
## Female Male  
## Heavy 0.02127660 0.02553191  
## Never 0.42127660 0.37872340  
## Occas 0.03829787 0.04255319  
## Regul 0.02127660 0.05106383

#the same as  
smoking\_men <- nrow(survey[survey$Sex == 'Male' & survey$Smoke == 'Regul', ])  
smoking\_men / nrow(survey)

## [1] 0.05485232

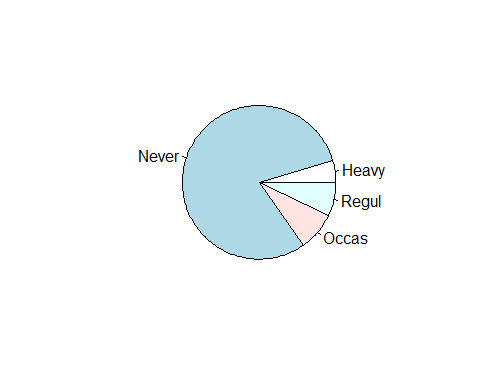
#случаен мъж да се окаже редовен пушач.Стойността на всяка клетка се дели на сумата  
#от редовете  
prop.table(table(survey$Sex, survey$Smoke), 1)

##   
## Heavy Never Occas Regul  
## Female 0.04237288 0.83898305 0.07627119 0.04237288  
## Male 0.05128205 0.76068376 0.08547009 0.10256410

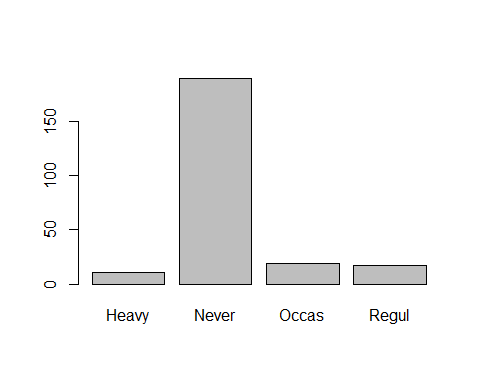
#случаен редовен пушач да се окаже мъж. Клетката се дели на сумата от колоните  
prop.table(table(survey$Sex, survey$Smoke), 2)

##   
## Heavy Never Occas Regul  
## Female 0.4545455 0.5265957 0.4736842 0.2941176  
## Male 0.5454545 0.4734043 0.5263158 0.7058824

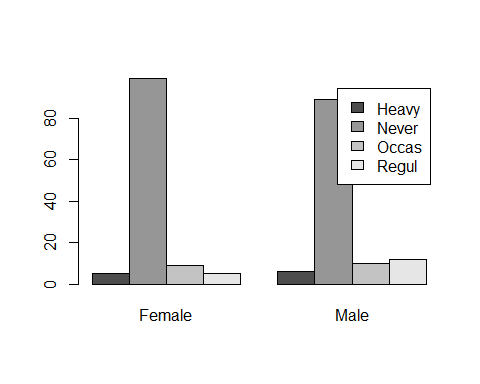
#2.  
#направете графики за пушачите и за пола  
  
#графики за пушенето  
pie(table(survey$Smoke))



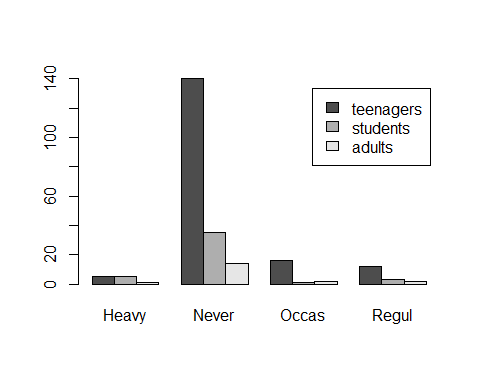
barplot(table(survey$Smoke))



#графика за пушенето и пола  
barplot(table(survey$Smoke, survey$Sex), beside = T, legend = T)



#3.  
#за да разделим някаква информация на интервали, които ние искаме ползваме cut  
groups <- cut(survey$Age, c(0, 20, 25, 100), c('teenagers', 'students', 'adults'))  
  
#правим го на графика  
table(groups, survey$Smoke) %>%  
 barplot(legend = T, beside = T)



#4.  
s <- sd(survey$Height, na.rm = T)  
  
med <- median(survey$Height,na.rm = T)  
  
m <- mean(survey$Height, na.rm = T)  
  
quantile(survey$Height, na.rm = T)

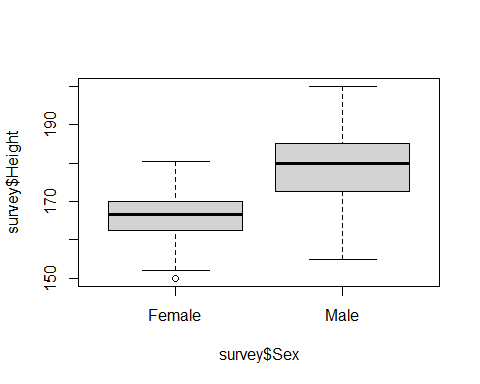
## 0% 25% 50% 75% 100%   
## 150 165 171 180 200

#брой различаващи се от средната височина с неповече от 2 стандартни отклонения  
cut(survey$Height, c(0, m - s, m + s, 300)) %>%  
 table()

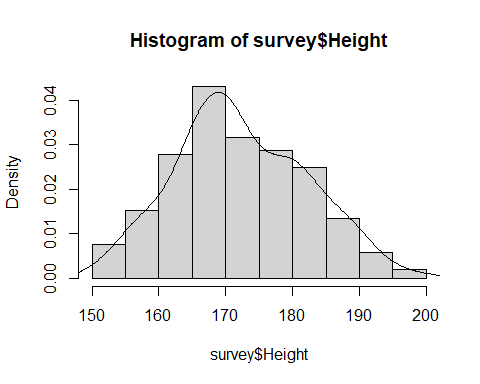
## .  
## (0,163] (163,182] (182,300]   
## 28 143 38

От упражение 3

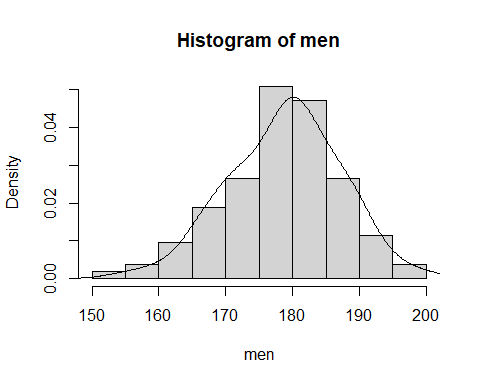
#1.  
#графика според височината и пола  
boxplot(survey$Height ~ survey$Sex)



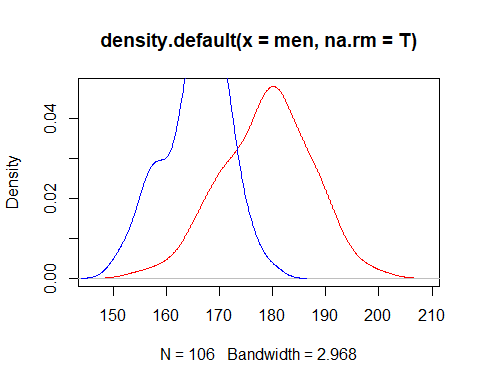
#хистограма според височината и имаме плътността  
hist(survey$Height, probability = T)  
 lines(density(survey$Height, na.rm = T))



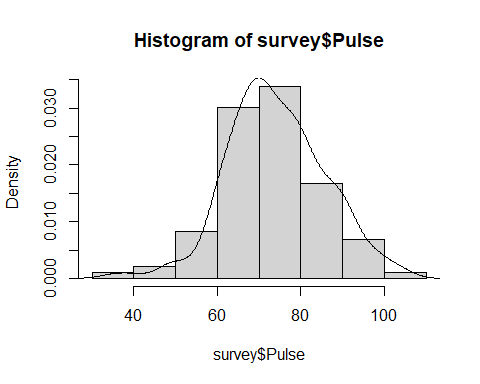
#хистограма според височината на мъжете и имаме линия за плътност  
men <- survey$Height[survey$Sex == 'Male']  
women <- survey$Height[survey$Sex == 'Female']  
  
hist(men, probability = T)  
 lines(density(men, na.rm = T))



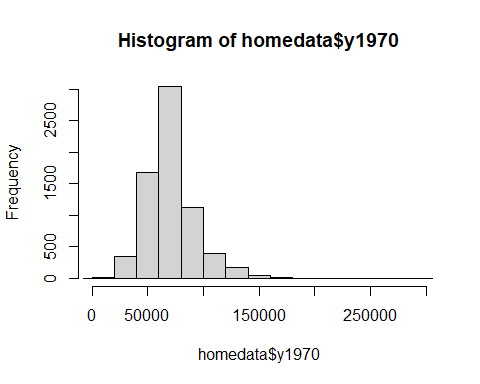
#графика за плътностите на височините на двата пола  
plot(density(men, na.rm = T), col='red')  
lines(density(women, na.rm = T), col='blue')



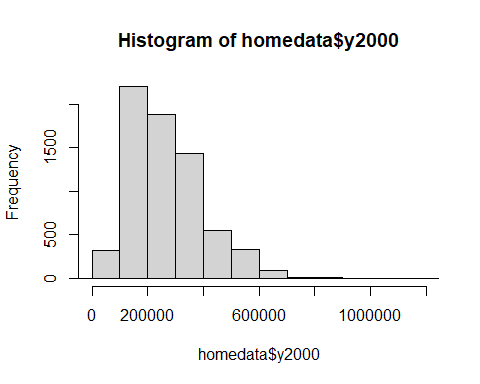
#2.Histogram for the pulse of the students including the density  
hist(survey$Pulse, probability = T)  
lines(density(survey$Pulse, na.rm = T))



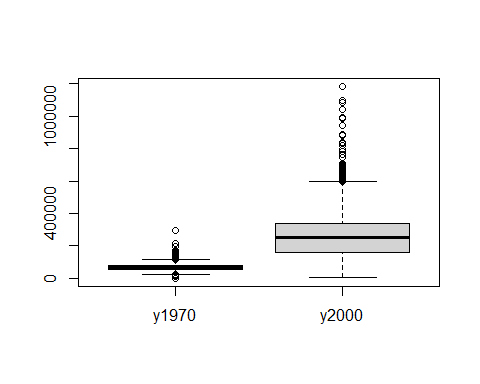
#3.  
#графиките за къщите от 1970 и 2000г.  
hist(homedata$y1970)  
lines(density(homedata$y1970, na.rm = T))



hist(homedata$y2000)  
lines(density(homedata$y2000, na.rm = T))



#сравняваме цените на къщите през 1970 и 2000г. и тяхната корелация  
boxplot(homedata)



correlation <- cor(homedata$y1970, homedata$y2000)

#4.  
#View(anscombe)  
  
#boxplot(anscombe)

От упражнение 4

dice = function(N = 100){  
 samples <- sample(1:6, size = 100, replace = TRUE)  
   
 result <- sum(samples == 6)  
   
 result  
}  
  
#емпирична вероятност  
dice() / 100

## [1] 0.15

birthdays = function(p = 0.5){  
   
 prob = 1  
 for(i in 1:365){  
 prob = prob \* (366 - i) / 365  
   
 if(prob < 1 - p) break  
 }  
 return(i)  
}  
  
birthdays()

## [1] 23

game\_one = function(father, mother){  
   
 wins = 0  
   
 for(i in 1:1000){  
 vs\_mom <- sample(0:1, 2, replace = T, prob = c(1 - mother, mother))  
 vs\_dad <- sample(0:1, 1, replace = T, prob = c(1- father, father))  
   
 if(vs\_mom[1] == 1 & vs\_dad == 1 | vs\_mom[2] == 1 & vs\_dad == 1){  
 wins = wins + 1  
 }  
   
 }  
 return(wins/1000)  
}  
  
game\_one(0.3, 0.4)

## [1] 0.184

#4.  
  
presents = function(n = 20){  
   
 for(j in 1:10000){  
 counter = 1  
   
 x <- sample(1:n, n, replace = FALSE)  
   
 for(i in 1:20){  
   
 if(i == x[i]){  
 counter = counter + 1  
 break  
 }  
 }  
 return(n - counter)  
 }  
}  
  
presents() / 10000

## [1] 0.0019

#5.  
  
coins = function(){  
 for(i in 1:10000){  
   
 x <- sample(0:1, 5, replace = T)  
   
 if(x[1] == 1 & x[2] == 1 & x[3] == 0 & x[4] == 1 & x[5] == 0){  
 break  
 }  
 }  
 return(i)  
}  
  
coins()

## [1] 88

От упражнение 5

#вероятността да се паднат по-малко от 5 шестици при хвърляне на 30 зара  
pbinom(q = 4, size = 30, prob = 1/6)

## [1] 0.4243389

#взимаме извадка от 10000 по 30 хвърляния на зар и го правим на таблица - това е емп. вер.  
thrown\_dices <- rbinom(n = 10000, size = 30, prob = 1/6)  
  
thrown\_dices %>% table() %>% prop.table()

## .  
## 0 1 2 3 4 5 6 7 8 9 10   
## 0.0045 0.0280 0.0722 0.1370 0.1852 0.1828 0.1612 0.1090 0.0672 0.0330 0.0135   
## 11 12 13 16   
## 0.0044 0.0014 0.0005 0.0001

#това е теоритичната вероятност  
dbinom(0:6, size = 30, prob = 1/6)

## [1] 0.00421272 0.02527632 0.07330133 0.13682915 0.18471936 0.19210813 0.16009011

#с вероятност 0,75 да се паднат повече от колко шестици  
#понеже нямаме ф-я за повече от ние ще променим твърдението с неговото обратно  
#понеже qbinom показва колко най-много шестици ще се паднат за някаква вероятност  
#тоест ние го променяме колко най-много ще се паднат за 0.25 вероятност  
qbinom(p = 0.25, size = 30, prob = 1/6)

## [1] 4

qbinom(p = 0.75, size = 30, prob = 1/6, lower.tail = FALSE)

## [1] 4

#2.  
  
#имаме пет неуспеха преди 3тия успех като вероятността за успех е 0.2  
# x е квантил  
dnbinom(x = 5, size = 3, prob = 0.2)

## [1] 0.05505024

#вероятност да са му нужни повече от 6 изтрела  
#q е бр. неуспехи  
pnbinom(q = 3, size = 3, prob = 0.2, lower.tail = FALSE)

## [1] 0.90112

#вероятност да му трябват между 5 и 8 изтрела вкл.  
pnbinom(q = 5, size = 3, prob = 0.2) - pnbinom(1,3,0.2)

## [1] 0.1758822

#3.  
balls = function(){  
   
 total = 13  
 white\_balls = 7  
 black\_balls = 6  
   
 num\_white = 0  
   
 for(i in 1:8){  
   
 white\_p = white\_balls / total  
 black\_p = black\_balls / total  
   
 x <- sample(c(0,1), size = 1, prob = c(black\_p, white\_p))  
   
 if(x == 0){  
 black\_balls = black\_balls - 1  
 }  
 else{  
 white\_balls = white\_balls - 1  
 num\_white = num\_white + 1  
 }  
   
 total = total - 1  
 }  
 num\_white  
}  
  
replicated <- replicate(1000, balls())  
  
  
mean(replicated)

## [1] 4.269

sd(replicated)

## [1] 0.9140429

min(replicated)

## [1] 2

max(replicated)

## [1] 7

sum(replicated == 3) / 1000

## [1] 0.175

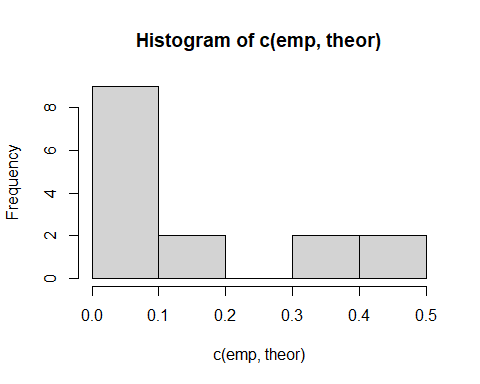
#емпирична вероятност  
emp <- replicated %>% table() %>% prop.table()  
  
#теоритична вероятност  
theor <- dhyper(0:8, 7, 6, 8)  
  
hist(c(emp, theor) , beside = T)

## Warning in plot.window(xlim, ylim, "", ...): "beside" is not a graphical  
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):  
## "beside" is not a graphical parameter

## Warning in axis(1, ...): "beside" is not a graphical parameter

## Warning in axis(2, ...): "beside" is not a graphical parameter

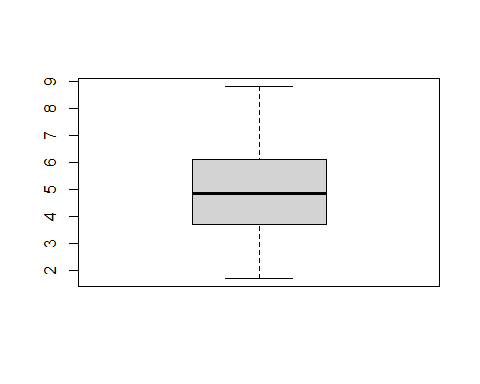


#4.  
n = 100  
  
dbinom(2, size = n, prob = 5/(2\*n))

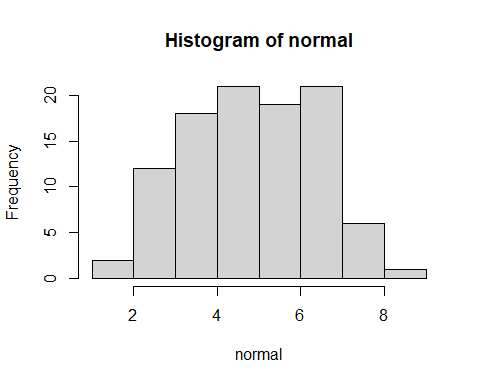
## [1] 0.2587841

От упражнение 6

#нормално разпр с боксплот и хистограма  
normal <- rnorm(100, 5, sqrt(2))  
  
boxplot(normal)



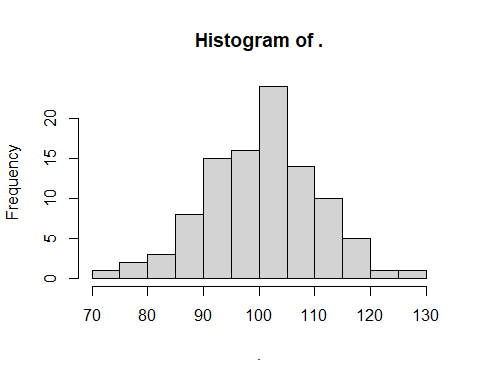
hist(normal)



#правим някаква извадка  
 s <- seq(1, 8, 0.2)  
   
#теоритична вероятност   
dnorm(s, 5, sqrt(2))

## [1] 0.005166746 0.007631185 0.011047931 0.015677760 0.021807265 0.029732572  
## [7] 0.039735427 0.052051997 0.066836087 0.084119899 0.103776874 0.125492144  
## [13] 0.148746447 0.172818715 0.196810858 0.219695645 0.240385325 0.257815227  
## [19] 0.271033697 0.279287902 0.282094792 0.279287902 0.271033697 0.257815227  
## [25] 0.240385325 0.219695645 0.196810858 0.172818715 0.148746447 0.125492144  
## [31] 0.103776874 0.084119899 0.066836087 0.052051997 0.039735427 0.029732572

#2.  
#n - брой сл. в  
#k - брой стойности, които ни дава всяко разпределение  
# fn - distribution function and ... is her arguments  
  
xsim <- function(n, k, fn, ...){  
   
 #вектор пълен с нули. В него се събират стойностите поиндексно на всяко разпределение  
 s <- rep(0, k)  
 for (i in 1:n) {  
 s <- s + fn(k, ... )  
 }  
 #връща се вектор със сумата поиндексно на всички разпределения  
 s  
}  
  
xsim(100, 100, rexp) %>% hist()



#това ни показва граничната теор - т.е при сумиране на независими еднакво разпр сл.в  
#че се получава нормално разпределение

#4.   
#пъпеши по-малки от 20 т.е трето качество  
small <- pnorm(20, 25, 6)  
  
#първата половина от по-големите  
medium <- (1 - small) / 2  
  
big <- medium  
  
#колко да е голям за да бъде трето качество   
qnorm(big + medium, 25, 6)

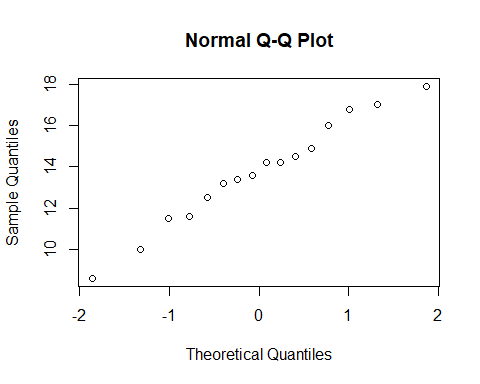
## [1] 30

От упражнение 7

#1. a)ако ни е известно станд отклонение  
n <- 20  
sd <- 2  
  
x <- rnorm(n, 3, sd)  
  
q <- qnorm(0.975, 3, 2)  
  
left\_interval <- mean(x) - q \* (sd / sqrt(n))  
  
right\_interval <- mean(x) + q \* (sd / sqrt(n))  
  
#b)ако не ни е известно станд отклонение  
  
n <- 20  
  
theor\_mean <- 3  
theor\_sd <- 2  
  
x <- rnorm(n, theor\_mean, theor\_sd)  
  
m <- mean(x)  
  
sd <- sd(x)  
  
q <- qt(p = 0.975, df = n - 1)  
  
left\_interval <- m - q \* sd / sqrt(n)  
  
right\_interval <- m + q \* sd / sqrt(n)  
  
#можем и така да намерим доверителния интервал ако знаем че х е нормално разпределено  
t.test(x)

##   
## One Sample t-test  
##   
## data: x  
## t = 6.8338, df = 19, p-value = 1.6e-06  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 2.276277 4.286187  
## sample estimates:  
## mean of x   
## 3.281232

#2.  
data1 <- c(10.0, 13.6, 13.2, 11.6, 12.5, 14.2, 14.9, 14.5, 13.4, 8.6, 11.5, 16.0, 14.2, 16.8, 17.9, 17.0)  
  
#проверяваме дали е нормално разпределена  
qqnorm(data1)



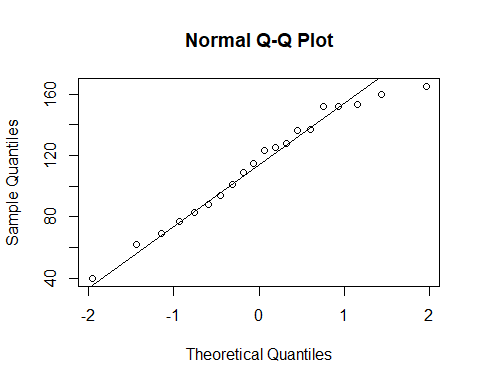
#правим доверителните интервали  
t.test(data1)

##   
## One Sample t-test  
##   
## data: data1  
## t = 21.65, df = 15, p-value = 9.976e-13  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 12.39066 15.09684  
## sample estimates:  
## mean of x   
## 13.74375

t.test(data1, conf.level = 0.90)

##   
## One Sample t-test  
##   
## data: data1  
## t = 21.65, df = 15, p-value = 9.976e-13  
## alternative hypothesis: true mean is not equal to 0  
## 90 percent confidence interval:  
## 12.63088 14.85662  
## sample estimates:  
## mean of x   
## 13.74375

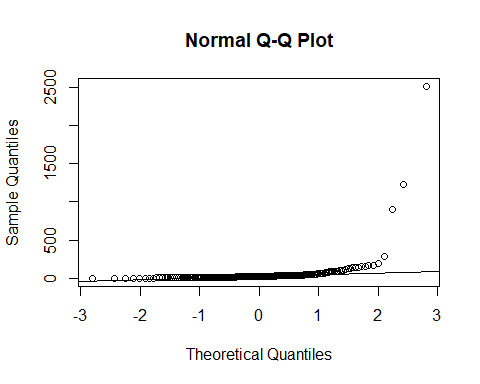
#3.a)  
qqnorm(rat)  
qqline(rat)



t.test(rat, conf.level = 0.96)

##   
## One Sample t-test  
##   
## data: rat  
## t = 14.176, df = 19, p-value = 1.48e-11  
## alternative hypothesis: true mean is not equal to 0  
## 96 percent confidence interval:  
## 95.80624 131.09376  
## sample estimates:  
## mean of x   
## 113.45

#b)  
#данните не са нормално разпределени затова ползваме wilcox.test  
qqnorm(exec.pay)  
qqline(exec.pay)



wilcox.test(exec.pay, conf.int = T, conf.level = 0.96)

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: exec.pay  
## V = 19306, p-value < 2.2e-16  
## alternative hypothesis: true location is not equal to 0  
## 96 percent confidence interval:  
## 25.99996 33.00003  
## sample estimates:  
## (pseudo)median   
## 29.00002

#v)

#4.когато имаме пропорции ползваме това  
prop.test(87, 150, conf.level = 0.92)

##   
## 1-sample proportions test with continuity correction  
##   
## data: 87 out of 150, null probability 0.5  
## X-squared = 3.5267, df = 1, p-value = 0.06039  
## alternative hypothesis: true p is not equal to 0.5  
## 92 percent confidence interval:  
## 0.5051991 0.6514474  
## sample estimates:  
## p   
## 0.58

#6.  
smoke\_men <- nrow(survey[survey$Sex == 'Male' & survey$Smoke == 'Never', ])  
  
all\_men <- nrow(survey[survey$Sex == 'Male', ])  
  
prop.test(smoke\_men, all\_men, conf.level = 0.90)

##   
## 1-sample proportions test with continuity correction  
##   
## data: smoke\_men out of all\_men, null probability 0.5  
## X-squared = 32.303, df = 1, p-value = 1.319e-08  
## alternative hypothesis: true p is not equal to 0.5  
## 90 percent confidence interval:  
## 0.6908187 0.8260629  
## sample estimates:  
## p   
## 0.7647059

От упражнение 8

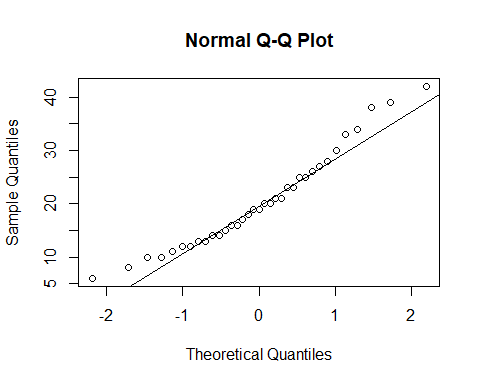
#1.check for norm distr  
# H0 - EX = 3  
data2 <- rnorm(100, mean = 2, sd = 2)  
  
#p-value< 0.05 отхвърляме хипотезата  
t.test(data2, mu = 3, alternative = 'two.sided')

##   
## One Sample t-test  
##   
## data: data2  
## t = -4.4363, df = 99, p-value = 2.376e-05  
## alternative hypothesis: true mean is not equal to 3  
## 95 percent confidence interval:  
## 1.646252 2.482987  
## sample estimates:  
## mean of x   
## 2.06462

t.test(data2, mu = 5, alternative = 'two.sided')

##   
## One Sample t-test  
##   
## data: data2  
## t = -13.922, df = 99, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 5  
## 95 percent confidence interval:  
## 1.646252 2.482987  
## sample estimates:  
## mean of x   
## 2.06462

#2.H0: дните за почивка да са 24 при п-жалуе > 0.2  
data3 <- vacation  
  
#seems normal distributed  
qqnorm(data3)  
qqline(data3)



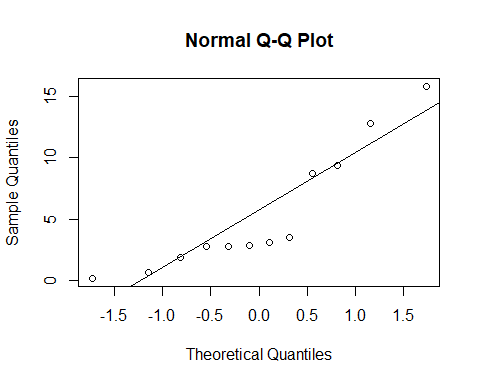
#p-value < 0,2 отхвърляме нулевата хипотеза  
t.test(vacation, mu = 24, alternative = 'two.sided')

##   
## One Sample t-test  
##   
## data: vacation  
## t = -2.2584, df = 34, p-value = 0.03045  
## alternative hypothesis: true mean is not equal to 24  
## 95 percent confidence interval:  
## 17.37768 23.65089  
## sample estimates:  
## mean of x   
## 20.51429

#3.H0 - 50 процента са доволни  
# H1 - < 50 процента са доволни  
  
prop.test(42, 100, p = 0.5, alternative = 'less')

##   
## 1-sample proportions test with continuity correction  
##   
## data: 42 out of 100, null probability 0.5  
## X-squared = 2.25, df = 1, p-value = 0.06681  
## alternative hypothesis: true p is less than 0.5  
## 95 percent confidence interval:  
## 0.0000000 0.5072341  
## sample estimates:  
## p   
## 0.42

#4.H0 - 5 minutes on the phone H1 - more than 5 mins  
data4 <- c(12.8, 3.5, 2.9, 9.4, 8.7, 0.7, 0.2, 2.8, 1.9, 2.8, 3.1, 15.8)  
  
qqnorm(data4)  
qqline(data4)



#not normal distribution  
shapiro.test(data4)

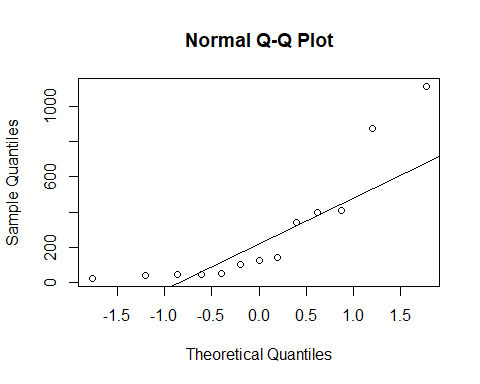
##   
## Shapiro-Wilk normality test  
##   
## data: data4  
## W = 0.83988, p-value = 0.0276

#we accept h0  
wilcox.test(data4, mu = 5, alternative = 'greater')

## Warning in wilcox.test.default(data4, mu = 5, alternative = "greater"): cannot  
## compute exact p-value with ties

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: data4  
## V = 39, p-value = 0.5156  
## alternative hypothesis: true location is greater than 5

#5.h0 - live more than 100 days h1- less than 100 days  
data5 <- cancer$stomach  
  
qqnorm(data5)  
qqline(data5)



#not normal distr  
shapiro.test(data5)

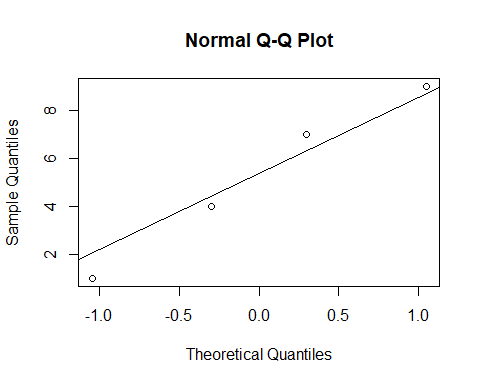
##   
## Shapiro-Wilk normality test  
##   
## data: data5  
## W = 0.75473, p-value = 0.002075

#we accept h0  
wilcox.test(data5, mu = 100, alternative = 'less')

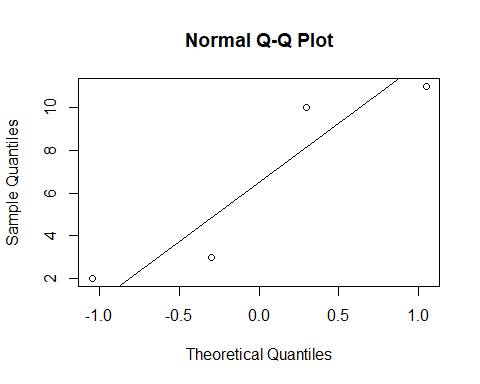
##   
## Wilcoxon signed rank exact test  
##   
## data: data5  
## V = 61, p-value = 0.8633  
## alternative hypothesis: true location is less than 100

От упражение 9

#1. H0 - equal h1 - not equal  
x <- c(4, 1, 7, 9)  
y <- c(10, 3, 2, 11)  
  
#не са норм разпр  
qqnorm(x)  
qqline(x)



qqnorm(y)  
qqline(y)



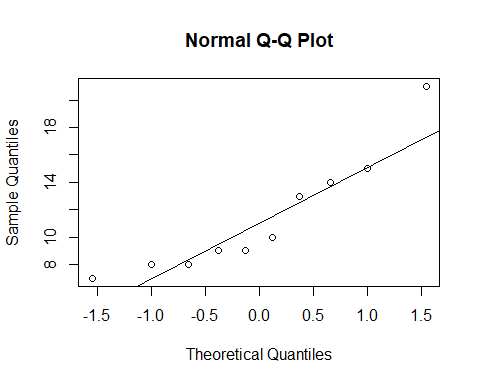
#приемаме нулевата хипотеза  
wilcox.test(x, y, alternative = 'two.sided')

##   
## Wilcoxon rank sum exact test  
##   
## data: x and y  
## W = 6, p-value = 0.6857  
## alternative hypothesis: true location shift is not equal to 0

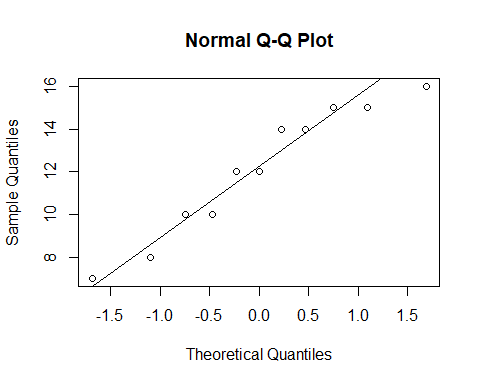
#2.H0 - equal H1 - greater  
  
#we accept h1  
prop.test(c(351, 71), c(605, 195), alternative = 'greater')

##   
## 2-sample test for equality of proportions with continuity correction  
##   
## data: c(351, 71) out of c(605, 195)  
## X-squared = 26.761, df = 1, p-value = 1.151e-07  
## alternative hypothesis: greater  
## 95 percent confidence interval:  
## 0.1470851 1.0000000  
## sample estimates:  
## prop 1 prop 2   
## 0.5801653 0.3641026

#3.H0 - less H1 - more  
  
before <- c( 15, 10, 13, 7, 9, 8, 21, 9, 14, 8)  
after <- c(15, 14, 12, 8, 14, 10, 7, 16, 10, 15, 12)  
  
#normal distr  
qqnorm(before)  
qqline(before)



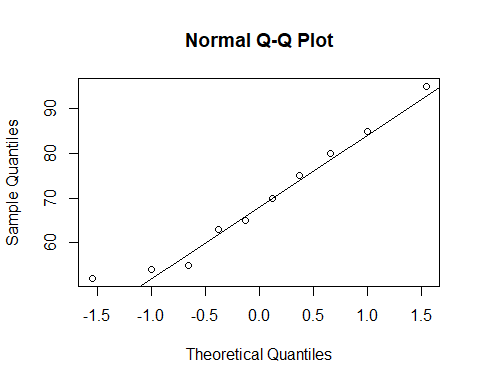
qqnorm(after)  
qqline(after)



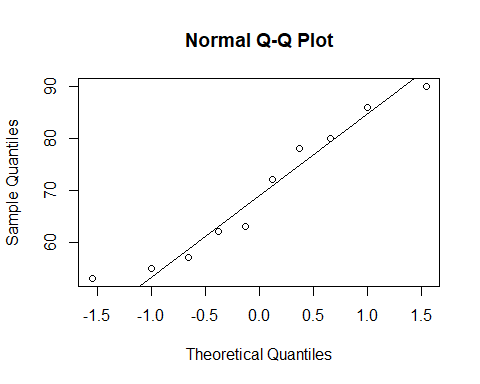
t.test(before, after, alternative = 'greater')

##   
## Welch Two Sample t-test  
##   
## data: before and after  
## t = -0.41894, df = 15.853, p-value = 0.6596  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## -3.571812 Inf  
## sample estimates:  
## mean of x mean of y   
## 11.40000 12.09091

#4.Х0 - че са равни Х1 - че са различни   
  
radar1 <- c(70, 85, 63, 54, 65, 80, 75, 95, 52, 55)  
  
radar2 <- c(72, 86, 62, 55, 63, 80, 78, 90, 53, 57)  
  
#they are normally distributed  
qqnorm(radar1)  
qqline(radar1)



qqnorm(radar2)  
qqline(radar2)



t.test(radar1, radar2, alternative = 'two.sided', paired = T)

##   
## Paired t-test  
##   
## data: radar1 and radar2  
## t = -0.26941, df = 9, p-value = 0.7937  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -1.879354 1.479354  
## sample estimates:  
## mean of the differences   
## -0.2

От упражнение 10

#1.H0 - data elements are equal H1- they are not  
  
data6 <- c(125, 410, 310, 300, 318, 298, 148)  
  
#we cant accept H0  
chisq.test(data6)

##   
## Chi-squared test for given probabilities  
##   
## data: data6  
## X-squared = 223.84, df = 6, p-value < 2.2e-16

#2.H0 - all digits have the same prob H1 - they don't   
  
#взимаме първите 200 цифри и виждаме колко често всяка една се среща  
data7 <- pi2000[1:200] %>% table()  
  
#правим хи-квадрат теста  
#можем да приемем нулевата хипотеза  
chisq.test(data7)

##   
## Chi-squared test for given probabilities  
##   
## data: data7  
## X-squared = 7.2, df = 9, p-value = 0.6163

#3.  
#теоритичните вероятности за срещането на буквите в англ език  
probs <- c(0.1270, 0.0956, 0.0817, 0.0751, 0.0697, 0.0675, 0.4834)  
  
#срещането на буквите от нашия текст  
letters <- c(102, 108, 90, 95, 82, 40, 519)  
  
#правим проверка дали са равни  
chisq.test(letters, p = probs)

##   
## Chi-squared test for given probabilities  
##   
## data: letters  
## X-squared = 26.396, df = 6, p-value = 0.0001878

#p-value е много малко затова отхвърляме хипотезата

#4.искаме да видим дали колан / без колан са независими сл.в Това става като подадед  
#на ф-ята матрица и тя прави проверката, т.е нулевата хипотеза е, че данните са независими  
# иначе са зависими  
  
belt <- c(12813, 647, 359, 42)  
  
nobelt <- c(65963, 4000, 2642, 303)  
  
data8 <- matrix(belt, nrow = 1, ncol = 4)  
  
data9 <- rbind(data8, nobelt)  
  
#p-value е много малко следователно отхвърляме нулевата хипотеза, т.е са зависими  
#колан влияе на нараняването при катастрофа  
chisq.test(data9)

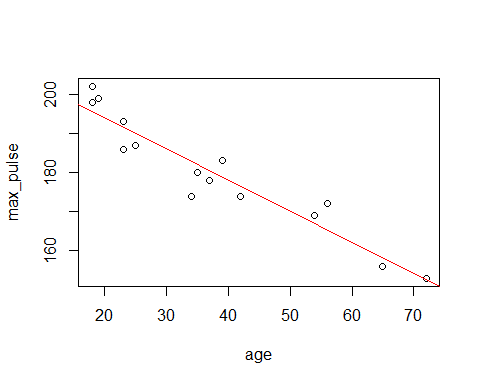
##   
## Pearson's Chi-squared test  
##   
## data: data9  
## X-squared = 59.224, df = 3, p-value = 8.61e-13

#5.  
  
mon <- c(44, 14, 15, 3)  
tues <- c(74, 25, 20, 5)  
wed <- c(79, 27, 20, 5)  
thurs <- c(72, 24, 23, 0)  
fri <- c(31, 10, 9, 0)  
  
final <- matrix(c(mon, tues, wed, thurs, fri), nrow = 5)  
  
#отхвърляме нулевата хипотеза че са независими , т.е има връзка между деня и качеството на стоката  
chisq.test(final)

##   
## Pearson's Chi-squared test  
##   
## data: final  
## X-squared = 350.71, df = 12, p-value < 2.2e-16

От упражнение 11

#1.Създаваме дата фрейм за данните  
patients\_df <- data.frame(  
 age = c(18, 23, 25, 35, 65, 54, 34, 56, 72, 19, 23, 42, 18, 39, 37),  
 max\_pulse = c(202, 186, 187, 180, 156, 169, 174, 172, 153, 199, 193, 174, 198, 183, 178)  
)  
  
#това е за модел на линейната регресия  
model1 <- lm(patients\_df$max\_pulse ~ patients\_df$age, data = patients\_df)  
  
  
plot(patients\_df)  
abline(model1, col = "red")



summ\_lm <- summary(model1)  
  
n <- nrow(patients\_df)  
  
# Тестване на хипотезата, че бета1 = -1  
# H0 :- "бета\_1 = -1"  
  
# Стандартно отклонение(грешка) на оценката за бета1  
std\_b1 <- summ\_lm$coefficients[2, 2]  
  
# оценката за бета1  
est\_b1 <- summ\_lm$coefficients[2, 1]  
  
  
# Параметър за бета1 под нулева хипотеза  
b1\_null\_hyp <- -1  
  
# Изграждане на т-статистика   
t\_statistic <- (est\_b1 - b1\_null\_hyp) / std\_b1  
  
# Вероятност да наблюдваме тази т-статистика (или по-крайна) при положение, че е вярна нулевата хипотеза  
pval <- 2 \* pt(t\_statistic, n - 2, lower.tail = FALSE)  
  
  
# Прогнозиране за възрасти 30, 40 и 50  
predict.lm(  
 model1,  
 newdata = data.frame(age = c(30, 40, 50)),  
 interval = "confidence",  
 level = 0.9  
)

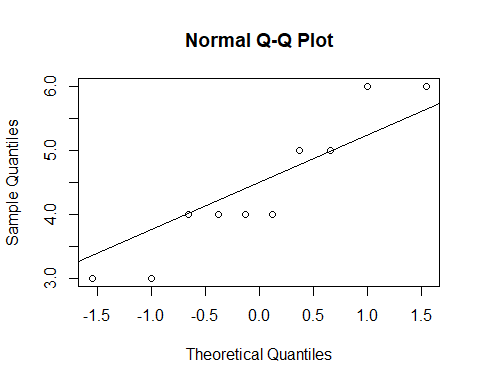
## Warning: 'newdata' had 3 rows but variables found have 15 rows

## fit lwr upr  
## 1 195.6894 192.5083 198.8705  
## 2 191.7007 188.9557 194.4458  
## 3 190.1053 187.5137 192.6969  
## 4 182.1280 180.0149 184.2411  
## 5 158.1962 154.1798 162.2127  
## 6 166.9712 164.0309 169.9116  
## 7 182.9258 180.7922 185.0593  
## 8 165.3758 162.2564 168.4952  
## 9 152.6121 147.8341 157.3902  
## 10 194.8917 191.8028 197.9805  
## 11 191.7007 188.9557 194.4458  
## 12 176.5439 174.3723 178.7155  
## 13 195.6894 192.5083 198.8705  
## 14 178.9371 176.8337 181.0405  
## 15 180.5326 178.4390 182.6262

Oт упражнение 12

От упражнение 13

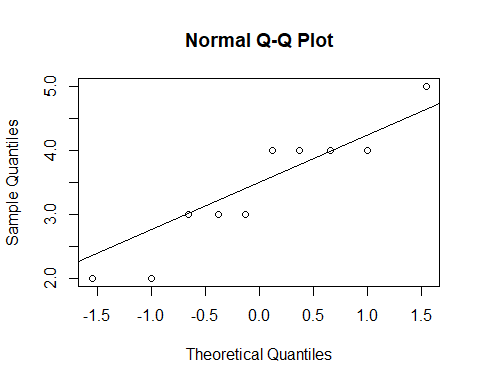
#нулевата хипотеза е че имат равни средни трите извадки от данни  
#правим дата фрейм с данните от задачата  
 exams\_df <- data.frame(  
examinor1 = c(5, 4, 4, 6, 4, 6, 3, 3, 4, 5),  
examinor2 = c(3, 2, 4, 5, 3, 4, 3, 4, 2, 4),  
examinor3 = c(4 ,6 ,4 ,2 ,4 ,5 ,5 ,3 ,6 ,4)  
)   
  
 stacked\_exam\_df <- stack(exams\_df)  
   
#гледаме дали са нормално разпределени данните  
   
qqnorm(exams\_df$examinor1)  
qqline(exams\_df$examinor1)



shapiro.test(exams\_df$examinor1)

##   
## Shapiro-Wilk normality test  
##   
## data: exams\_df$examinor1  
## W = 0.89165, p-value = 0.177

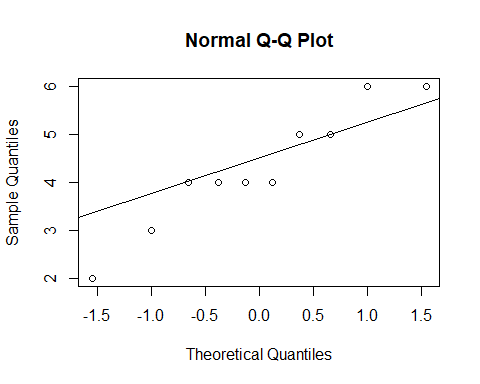
qqnorm(exams\_df$examinor2)  
qqline(exams\_df$examinor2)



shapiro.test(exams\_df$examinor2)

##   
## Shapiro-Wilk normality test  
##   
## data: exams\_df$examinor2  
## W = 0.90444, p-value = 0.2449

qqnorm(exams\_df$examinor3)  
qqline(exams\_df$examinor3)



shapiro.test(exams\_df$examinor3)

##   
## Shapiro-Wilk normality test  
##   
## data: exams\_df$examinor3  
## W = 0.92883, p-value = 0.4365

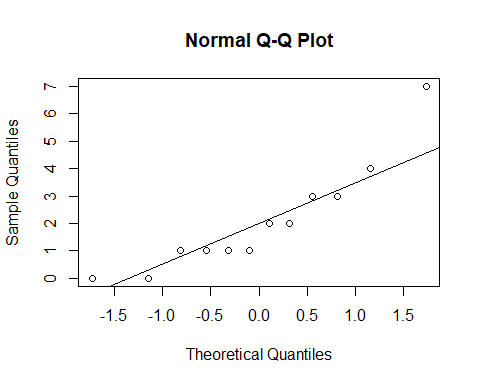
#и трите са нормално разпределени тоест можем да направим тест дали имат еднакво средно ако са норм разпр  
  
oneway.test(values ~ ind, data = stacked\_exam\_df)

##   
## One-way analysis of means (not assuming equal variances)  
##   
## data: values and ind  
## F = 2.7825, num df = 2.000, denom df = 17.811, p-value = 0.0888

#Не можем да отхвърлим хипотезата че имат равни средни  
#друг начин да се провери съшата хипотеза  
  
anova(lm(values ~ ind, data = stacked\_exam\_df))

## Analysis of Variance Table  
##   
## Response: values  
## Df Sum Sq Mean Sq F value Pr(>F)  
## ind 2 6.067 3.0333 2.4894 0.1018  
## Residuals 27 32.900 1.2185

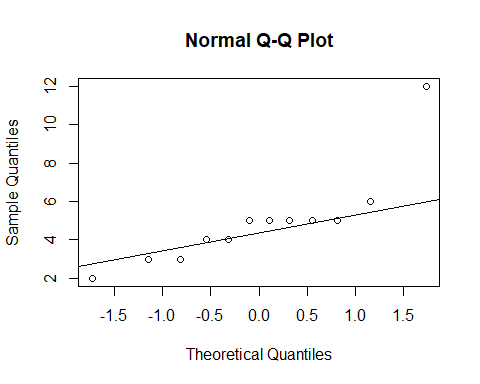
#2.  
  
groupC <- InsectSprays$count[InsectSprays$spray == 'C']  
  
groupD <- InsectSprays$count[InsectSprays$spray == 'D']  
  
groupE <- InsectSprays$count[InsectSprays$spray == 'E']  
  
#изглежда ми сравнително нормално разпределени  
qqnorm(groupC)  
qqline(groupC)



shapiro.test(groupC)

##   
## Shapiro-Wilk normality test  
##   
## data: groupC  
## W = 0.85907, p-value = 0.04759

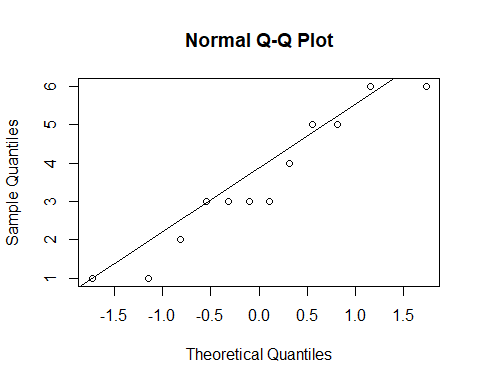
qqnorm(groupD)  
qqline(groupD)



shapiro.test(groupD)

##   
## Shapiro-Wilk normality test  
##   
## data: groupD  
## W = 0.75063, p-value = 0.002713

qqnorm(groupE)  
qqline(groupE)



shapiro.test(groupE)

##   
## Shapiro-Wilk normality test  
##   
## data: groupE  
## W = 0.92128, p-value = 0.2967

#p-value-то е много малко следователно можем да твърдим че някои от препаратите действат по-добре от други  
oneway.test(count ~ spray, data = InsectSprays)

##   
## One-way analysis of means (not assuming equal variances)  
##   
## data: count and spray  
## F = 36.065, num df = 5.000, denom df = 30.043, p-value = 7.999e-12

#3.взимаме данните от файла  
drug\_df <- read.csv("./data.txt")  
  
#тъй като имаме сдвоени данни, т.е даваме лекартво на един и същ пациент ползваме aov  
aov(response ~ drug + Error(patient), data = drug\_df) %>% summary()

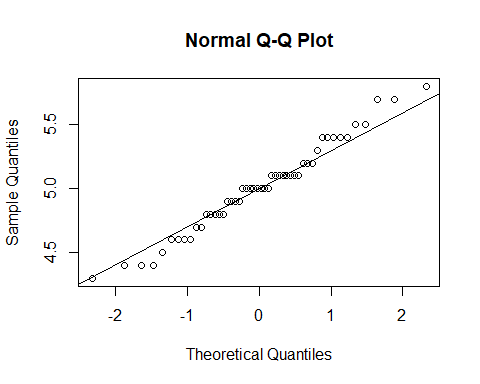
##   
## Error: patient  
## Df Sum Sq Mean Sq F value Pr(>F)  
## Residuals 1 72.9 72.9   
##   
## Error: Within  
## Df Sum Sq Mean Sq F value Pr(>F)  
## drug 1 36 36.00 0.565 0.462  
## Residuals 17 1082 63.66

#не можем да отхвърлим хопотезата, че имаме лекарствата действат еднакво

#4.  
iris

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa  
## 7 4.6 3.4 1.4 0.3 setosa  
## 8 5.0 3.4 1.5 0.2 setosa  
## 9 4.4 2.9 1.4 0.2 setosa  
## 10 4.9 3.1 1.5 0.1 setosa  
## 11 5.4 3.7 1.5 0.2 setosa  
## 12 4.8 3.4 1.6 0.2 setosa  
## 13 4.8 3.0 1.4 0.1 setosa  
## 14 4.3 3.0 1.1 0.1 setosa  
## 15 5.8 4.0 1.2 0.2 setosa  
## 16 5.7 4.4 1.5 0.4 setosa  
## 17 5.4 3.9 1.3 0.4 setosa  
## 18 5.1 3.5 1.4 0.3 setosa  
## 19 5.7 3.8 1.7 0.3 setosa  
## 20 5.1 3.8 1.5 0.3 setosa  
## 21 5.4 3.4 1.7 0.2 setosa  
## 22 5.1 3.7 1.5 0.4 setosa  
## 23 4.6 3.6 1.0 0.2 setosa  
## 24 5.1 3.3 1.7 0.5 setosa  
## 25 4.8 3.4 1.9 0.2 setosa  
## 26 5.0 3.0 1.6 0.2 setosa  
## 27 5.0 3.4 1.6 0.4 setosa  
## 28 5.2 3.5 1.5 0.2 setosa  
## 29 5.2 3.4 1.4 0.2 setosa  
## 30 4.7 3.2 1.6 0.2 setosa  
## 31 4.8 3.1 1.6 0.2 setosa  
## 32 5.4 3.4 1.5 0.4 setosa  
## 33 5.2 4.1 1.5 0.1 setosa  
## 34 5.5 4.2 1.4 0.2 setosa  
## 35 4.9 3.1 1.5 0.2 setosa  
## 36 5.0 3.2 1.2 0.2 setosa  
## 37 5.5 3.5 1.3 0.2 setosa  
## 38 4.9 3.6 1.4 0.1 setosa  
## 39 4.4 3.0 1.3 0.2 setosa  
## 40 5.1 3.4 1.5 0.2 setosa  
## 41 5.0 3.5 1.3 0.3 setosa  
## 42 4.5 2.3 1.3 0.3 setosa  
## 43 4.4 3.2 1.3 0.2 setosa  
## 44 5.0 3.5 1.6 0.6 setosa  
## 45 5.1 3.8 1.9 0.4 setosa  
## 46 4.8 3.0 1.4 0.3 setosa  
## 47 5.1 3.8 1.6 0.2 setosa  
## 48 4.6 3.2 1.4 0.2 setosa  
## 49 5.3 3.7 1.5 0.2 setosa  
## 50 5.0 3.3 1.4 0.2 setosa  
## 51 7.0 3.2 4.7 1.4 versicolor  
## 52 6.4 3.2 4.5 1.5 versicolor  
## 53 6.9 3.1 4.9 1.5 versicolor  
## 54 5.5 2.3 4.0 1.3 versicolor  
## 55 6.5 2.8 4.6 1.5 versicolor  
## 56 5.7 2.8 4.5 1.3 versicolor  
## 57 6.3 3.3 4.7 1.6 versicolor  
## 58 4.9 2.4 3.3 1.0 versicolor  
## 59 6.6 2.9 4.6 1.3 versicolor  
## 60 5.2 2.7 3.9 1.4 versicolor  
## 61 5.0 2.0 3.5 1.0 versicolor  
## 62 5.9 3.0 4.2 1.5 versicolor  
## 63 6.0 2.2 4.0 1.0 versicolor  
## 64 6.1 2.9 4.7 1.4 versicolor  
## 65 5.6 2.9 3.6 1.3 versicolor  
## 66 6.7 3.1 4.4 1.4 versicolor  
## 67 5.6 3.0 4.5 1.5 versicolor  
## 68 5.8 2.7 4.1 1.0 versicolor  
## 69 6.2 2.2 4.5 1.5 versicolor  
## 70 5.6 2.5 3.9 1.1 versicolor  
## 71 5.9 3.2 4.8 1.8 versicolor  
## 72 6.1 2.8 4.0 1.3 versicolor  
## 73 6.3 2.5 4.9 1.5 versicolor  
## 74 6.1 2.8 4.7 1.2 versicolor  
## 75 6.4 2.9 4.3 1.3 versicolor  
## 76 6.6 3.0 4.4 1.4 versicolor  
## 77 6.8 2.8 4.8 1.4 versicolor  
## 78 6.7 3.0 5.0 1.7 versicolor  
## 79 6.0 2.9 4.5 1.5 versicolor  
## 80 5.7 2.6 3.5 1.0 versicolor  
## 81 5.5 2.4 3.8 1.1 versicolor  
## 82 5.5 2.4 3.7 1.0 versicolor  
## 83 5.8 2.7 3.9 1.2 versicolor  
## 84 6.0 2.7 5.1 1.6 versicolor  
## 85 5.4 3.0 4.5 1.5 versicolor  
## 86 6.0 3.4 4.5 1.6 versicolor  
## 87 6.7 3.1 4.7 1.5 versicolor  
## 88 6.3 2.3 4.4 1.3 versicolor  
## 89 5.6 3.0 4.1 1.3 versicolor  
## 90 5.5 2.5 4.0 1.3 versicolor  
## 91 5.5 2.6 4.4 1.2 versicolor  
## 92 6.1 3.0 4.6 1.4 versicolor  
## 93 5.8 2.6 4.0 1.2 versicolor  
## 94 5.0 2.3 3.3 1.0 versicolor  
## 95 5.6 2.7 4.2 1.3 versicolor  
## 96 5.7 3.0 4.2 1.2 versicolor  
## 97 5.7 2.9 4.2 1.3 versicolor  
## 98 6.2 2.9 4.3 1.3 versicolor  
## 99 5.1 2.5 3.0 1.1 versicolor  
## 100 5.7 2.8 4.1 1.3 versicolor  
## 101 6.3 3.3 6.0 2.5 virginica  
## 102 5.8 2.7 5.1 1.9 virginica  
## 103 7.1 3.0 5.9 2.1 virginica  
## 104 6.3 2.9 5.6 1.8 virginica  
## 105 6.5 3.0 5.8 2.2 virginica  
## 106 7.6 3.0 6.6 2.1 virginica  
## 107 4.9 2.5 4.5 1.7 virginica  
## 108 7.3 2.9 6.3 1.8 virginica  
## 109 6.7 2.5 5.8 1.8 virginica  
## 110 7.2 3.6 6.1 2.5 virginica  
## 111 6.5 3.2 5.1 2.0 virginica  
## 112 6.4 2.7 5.3 1.9 virginica  
## 113 6.8 3.0 5.5 2.1 virginica  
## 114 5.7 2.5 5.0 2.0 virginica  
## 115 5.8 2.8 5.1 2.4 virginica  
## 116 6.4 3.2 5.3 2.3 virginica  
## 117 6.5 3.0 5.5 1.8 virginica  
## 118 7.7 3.8 6.7 2.2 virginica  
## 119 7.7 2.6 6.9 2.3 virginica  
## 120 6.0 2.2 5.0 1.5 virginica  
## 121 6.9 3.2 5.7 2.3 virginica  
## 122 5.6 2.8 4.9 2.0 virginica  
## 123 7.7 2.8 6.7 2.0 virginica  
## 124 6.3 2.7 4.9 1.8 virginica  
## 125 6.7 3.3 5.7 2.1 virginica  
## 126 7.2 3.2 6.0 1.8 virginica  
## 127 6.2 2.8 4.8 1.8 virginica  
## 128 6.1 3.0 4.9 1.8 virginica  
## 129 6.4 2.8 5.6 2.1 virginica  
## 130 7.2 3.0 5.8 1.6 virginica  
## 131 7.4 2.8 6.1 1.9 virginica  
## 132 7.9 3.8 6.4 2.0 virginica  
## 133 6.4 2.8 5.6 2.2 virginica  
## 134 6.3 2.8 5.1 1.5 virginica  
## 135 6.1 2.6 5.6 1.4 virginica  
## 136 7.7 3.0 6.1 2.3 virginica  
## 137 6.3 3.4 5.6 2.4 virginica  
## 138 6.4 3.1 5.5 1.8 virginica  
## 139 6.0 3.0 4.8 1.8 virginica  
## 140 6.9 3.1 5.4 2.1 virginica  
## 141 6.7 3.1 5.6 2.4 virginica  
## 142 6.9 3.1 5.1 2.3 virginica  
## 143 5.8 2.7 5.1 1.9 virginica  
## 144 6.8 3.2 5.9 2.3 virginica  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

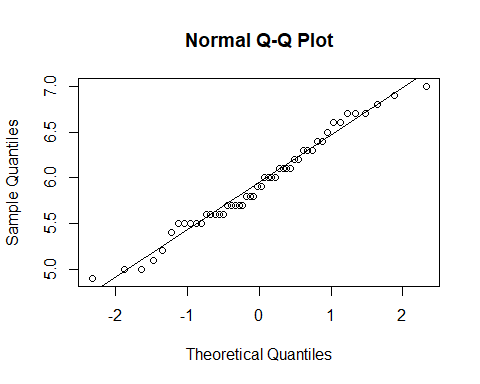
sort1 <- iris$Sepal.Length[iris$Species == 'setosa']  
  
sort2 <- iris$Sepal.Length[iris$Species == 'versicolor']  
  
sort3 <- iris$Sepal.Length[iris$Species == 'virginica']  
  
#checking whether the data is normal distributed  
qqnorm(sort1)  
qqline(sort1)



shapiro.test(sort1)

##   
## Shapiro-Wilk normality test  
##   
## data: sort1  
## W = 0.9777, p-value = 0.4595

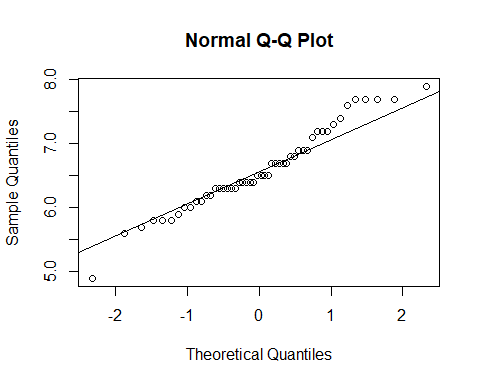
qqnorm(sort2)  
qqline(sort2)



shapiro.test(sort2)

##   
## Shapiro-Wilk normality test  
##   
## data: sort2  
## W = 0.97784, p-value = 0.4647

qqnorm(sort3)  
qqline(sort3)



shapiro.test(sort3)

##   
## Shapiro-Wilk normality test  
##   
## data: sort3  
## W = 0.97118, p-value = 0.2583

#all three are normally dirstributed  
  
# формула на модела (имаме два отклика)  
(cbind(iris$Sepal.Length, iris$Sepal.Width) ~ Species) %>%  
 # изпълнение на anova с много у променливи  
 manova(data = iris) %>%  
 # Обобщение  
 summary()

## Df Pillai approx F num Df den Df Pr(>F)   
## Species 2 0.94531 65.878 4 294 < 2.2e-16 \*\*\*  
## Residuals 147   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Извод: Различните сортове играят роля за размера на чашелистчетата  
 # някоя от групите има значително различно средно от останалите

От изпит 2017

#1.  
#number of people younger than 20 yrs  
length(Aids2$age[Aids2$age < 20])

## [1] 39

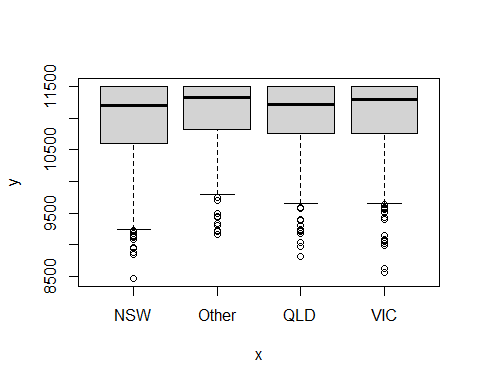
#sex of the patients with earliest diagnosis  
Aids2$sex[head(order(Aids2$diag), 5)]

## [1] M M M M M  
## Levels: F M

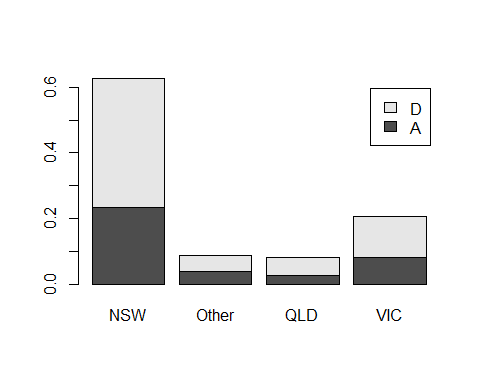
#men who got aids from blood  
men\_blood <- sum(Aids2$sex[Aids2$T.categ == 'blood'] == 'M')  
  
all\_men <- sum(Aids2$sex == 'M')  
  
men\_blood / all\_men

## [1] 0.02069717

#графика за щатът на пациента и смъртността  
plot(Aids2$state, Aids2$death, na.rm = T)



table(Aids2$status, Aids2$state) %>% prop.table() %>% barplot(legend.text = T)



?Aids2

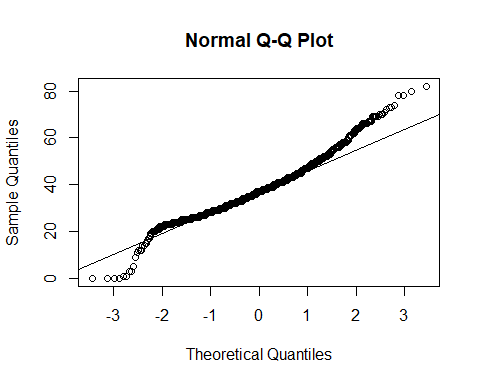
## starting httpd help server ... done

#2.  
#total women and women that died  
women <- sum(Aids2$sex == 'F')  
dead\_women <- sum(Aids2$status[Aids2$sex == 'F'] == 'D')  
  
men <- sum(Aids2$sex == 'M')  
dead\_men <- sum(Aids2$status[Aids2$sex == 'M'] == 'D')  
  
#Х0 - жените умират по-малко Х1 - умират повече  
prop.test(c(dead\_women, dead\_men), c(women, men), alternative = 'greater')

##   
## 2-sample test for equality of proportions with continuity correction  
##   
## data: c(dead\_women, dead\_men) out of c(women, men)  
## X-squared = 0.13041, df = 1, p-value = 0.641  
## alternative hypothesis: greater  
## 95 percent confidence interval:  
## -0.1173963 1.0000000  
## sample estimates:  
## prop 1 prop 2   
## 0.5955056 0.6201888

#можем да приемем нулевата хипотеза

number\_dead <- Aids2$age[Aids2$status == 'D']  
  
qqnorm(number\_dead)  
qqline(number\_dead)



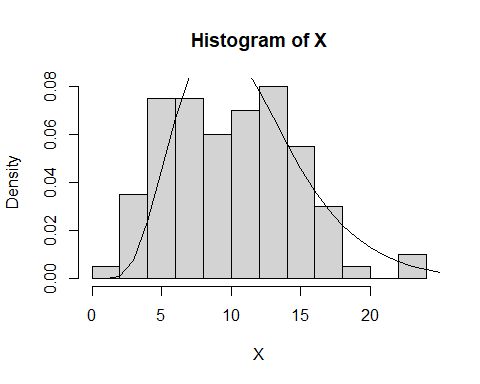
shapiro.test(number\_dead)

##   
## Shapiro-Wilk normality test  
##   
## data: number\_dead  
## W = 0.96911, p-value < 2.2e-16

#not normally distributed  
  
#Х0 - средната възраст е 38 Х1 - не е   
#приемаме h1 хипотеза  
wilcox.test(number\_dead, mu = 38, alternative = 'two.sided')

##   
## Wilcoxon signed rank test with continuity correction  
##   
## data: number\_dead  
## V = 649726, p-value = 0.00164  
## alternative hypothesis: true location is not equal to 38

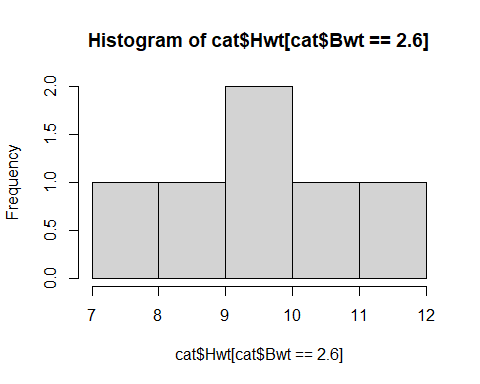
#4.  
X <- rchisq(100, df = 10)  
  
hist(X, probability = T)  
  
lines(dchisq(0:30, df = 10))



#5.  
cat <- cats[cats$Sex == 'M', ]  
  
#create the model  
s <- lm(Hwt ~ Bwt, data = cat) %>% summary()  
#based on the model we get that the heart and body weight are not independant   
#демек са зависими и сърцето се повлиява от теглото на котката  
  
t <- (s$coefficients[2, 1] - 5) / s$coefficients[2, 2]  
#t is negative => we calculate p-value like this:  
#df is equal to: numOfObservations - numOfEvaluatedParameters - 1  
# so: 97 - 2 - 1 = 94  
#Вярно ли е, че при котки по тежки с 1 кг сърцето е по тежко с 5 гр - H0  
pval = 2 \* pt(t, df = 94)  
#тук имаме p-value < 0.05 което значи че отхвърляме нулевата хипотеза, т.е горното не е вярно  
  
# pval = 2 \* pt(t, df = 94, lower.tail = F) if t > 0 only lower tail  
  
#we test if the distribution is normal so we can do t.test to get the conf.interval  
shapiro.test(cat$Hwt[cat$Bwt == 2.6])

##   
## Shapiro-Wilk normality test  
##   
## data: cat$Hwt[cat$Bwt == 2.6]  
## W = 0.96653, p-value = 0.8683

hist(cat$Hwt[cat$Bwt == 2.6])



#it is norm dist so we use t.test else we use wilcox.test()  
t.test(cat$Hwt[cat$Bwt == 2.6], conf.level = 0.95)

##   
## One Sample t-test  
##   
## data: cat$Hwt[cat$Bwt == 2.6]  
## t = 16.654, df = 5, p-value = 1.426e-05  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 8.005474 10.927859  
## sample estimates:  
## mean of x   
## 9.466667

От примерен тест 2017

#1.  
qnorm(p = 0.05)

## [1] -1.644854

#2.  
nrow(state.x77)

## [1] 50

#подреждаме щатите по ниво на необразованост  
dumb\_states <- head(order(state.x77[,3], decreasing = T), 5)  
#взимаме ги според индексите на първите 5 щата  
state.x77[dumb\_states, 3]

## Louisiana Mississippi South Carolina New Mexico Texas   
## 2.8 2.4 2.3 2.2 2.2

#states with life expectancy over 70  
old\_states <- state.x77[1:50, 4] > 70  
length(state.x77[old\_states, 4])

## [1] 41

#щат с най-голяма гъстота на населението  
pop <- state.x77[1:50,1]  
  
land <- state.x77[1:50, 8]  
  
density <- pop / land  
  
density[head(order(density, decreasing = T), 1)]

## New Jersey   
## 0.9750033

#общото население на петте най-големи щати  
biggest\_states <- head(order(state.x77[1:50, 8], decreasing = T), 5)  
sum(state.x77[biggest\_states, 1])

## [1] 35690

#3.ho - има по-малко подобрили се жени х1- има повече подобрили се мъже  
women <- 200  
men <- 100  
  
not\_accepted\_women <- women \* 38 / 100  
not\_accepted\_men <- men \* 50 / 100  
  
#приемаме хипотезата че е по-ефективно при жените отколкото при мъжете  
prop.test(c(not\_accepted\_women, not\_accepted\_men), c(women, men), alternative = 'greater' )

##   
## 2-sample test for equality of proportions with continuity correction  
##   
## data: c(not\_accepted\_women, not\_accepted\_men) out of c(women, men)  
## X-squared = 3.4637, df = 1, p-value = 0.9686  
## alternative hypothesis: greater  
## 95 percent confidence interval:  
## -0.2272546 1.0000000  
## sample estimates:  
## prop 1 prop 2   
## 0.38 0.50

#4.  
 data <- data.frame(  
anscombe$x3,  
anscombe$x4)  
  
l <- lm(data$anscombe.x3 ~ data$anscombe.x4, data = data)   
  
plot(data$anscombe.x3, data$anscombe.x4)  
abline(l, col = "red ", lwd = 2)

